Clustering methods- Simulation Design

Saritha Kodikara

February 18, 2022

Contents

1 Simulation Design

To evaluate the clustering methods for identifying taxa profiles that evolve similarly across time, we need to summarize the profiles across different subjects to reduce the subject dimension in the data. The most common way to reduce the subject dimension is to calculate the mean at each time point for each taxa (Straube et al. 2015). Straube et al. (2015) showed that the linear mixed model splines (LMMS) modelled profiles better discriminate between temporary changes when compared to other methods such as mean profiles and smoothing splines mixed-effects modelled profiles. In our simulation, we used LMMS data to assess the clustering performance.

We generated two-hundred reference time profiles from time one to time nine and were assigned to four clusters (50 profiles each). These reference profiles were then used to simulate five new profiles (corresponding to different individuals) with a fixed level of noise. For a given noise level, 100 data sets were generated with 200 time profiles and 5 individuals. The time profiles were then modeled with LMMS, resulting in 100 data sets of size (9 × 200) for each level of noise. Then these LMMS profiles were used as inputs in clustering methods and calculated the clustering accuracy by dividing the number of correctly classified profiles by the total number of time profiles. Three noise levels (0.5, 1.5, 3) were considered to assess the effect of inter-individual variability on clustering accuracy. We also used centered LMMS profiles, scaled LMMS profiles and centered and scaled LMMS profiles as inputs in each of the clustering methods.

```
#Simulation function
#https://github.com/abodein/timeOmics_frontiers/blob/master/Examples/Simulation.Rmd
generate_LMMS_data <- function(raw_data, N_Ind, noise){
  data.gather <- raw_data %>% as.data.frame() %>% rownames_to_column("time") %>%
    gather(feature, value, -time)
  for(ind in 1:N_Ind){
    vect <- vector(length = nrow(data.gather), mode = "numeric")
    for(x in 1:length(vect)){
      vect[x] <- rnorm(1, mean = data.gather$value[x], sd = noise)
    }
    names.tmp <- colnames(data.gather)
    data.gather <- data.frame(data.gather, vect)
    colnames(data.gather) <- c(names.tmp, LETTERS[ind])
  }
  sim_data <- data.gather %>% dplyr::select(-c(value)) %>%
    gather(ind, value, -c(time, feature)) %>%
    mutate(sample = paste0(ind, "_", time)) %>%
    dplyr::select(feature, value, sample) %>%
    spread(feature, value) %>%
    column_to_rownames("sample") %>%
    as.matrix()
  return(sim_data)
}


sim_lmm<-function(rawData,noise, nInd){
  s1<-generate_LMMS_data(rawData,nInd,noise)
  time <- rep(1:9, nInd)

  lmms.output <- lmms::lmmSpline(data = s1, time = time,
                                 sampleID = rownames(s1), deri = FALSE,
                                 basis = "p-spline", numCores = 4, timePredict = 1:9,
                                 keepModels = TRUE)
  modelled.data <- t(slot(lmms.output, 'predSpline'))
  return(modelled.data)
}
```

1.1 Reference Profiles

```
set.seed(1234)
# RAW DATA
c1.0 <- c(0, 0.5,1,1.1,1.2,1.8,2.5,5,9)
c2.0<-c(-1, 8, -1, 0, 0.5, -2.5, -3, 2, 2)
c3.0 <-  c(-2,4, 8, 6,4.5,4,3.9, 3, 1)
c4.0 <- c(2, -5, 1, -4, 0.5, -3.5, 0, -3,0.5)

lst <-list()
lst[["c1.0"]]<-c1.0
lst[["c2.0"]]<-c2.0
lst[["c3.0"]]<-c3.0
lst[["c4.0"]]<-c4.0

for(i in 1:49){
  a<-round(rnorm(1,2,1),1)
  b<-round(rnorm(1,2,1),1)
  z1<-(c1.0+a)*abs(b+1)
  z2<-(c2.0+a)*abs(b+1)
  z3<-(c3.0+a)*abs(b+1)
  z4<-(c4.0+a)*abs(b+1)
  lst[[paste0("c1.", i)]]<-assign(paste0("c1.", i),z1)
  lst[[paste0("c2.", i)]]<-assign(paste0("c2.", i),z2)
  lst[[paste0("c3.", i)]]<-assign(paste0("c3.", i),z3)
  lst[[paste0("c4.", i)]]<-assign(paste0("c4.", i),z4)
}
raw.data<-lst[order(names(lst))]
```

1.2 Data Simulation

In this section, we simulate data with noise level equals to 0.5, 1.3 and 3.

```
rawData<-raw.data
nInd<-5
noise<-c(0.5,1.5,3)
n_iter <- 100 # Number of iterations of the loop

for (i in noise){
  list_of_frames <- replicate(n_iter, data.frame())
  for(j in 1:n_iter) {
    df_Scenario1<-sim_lmm(rawData,i,nInd)
    list_of_frames[[j]] <-df_Scenario1
  }
  saveRDS(list_of_frames,file =  paste0("Data/clusData_",i,".Rdata"))
}
```

```
plotFunction<-function(clusData, plotTitle){
# gather data
data.gathered <- clusData %>% as.data.frame() %>%
  rownames_to_column("time") %>%
  mutate(time = as.numeric(time)) %>%
  pivot_longer(names_to="feature", values_to = 'value', -time)%>%
  mutate(Cluster= substr(feature, start = 1, stop = 2))

# plot profiles
ggplot(data.gathered,
       aes(x = time, y = value, group = feature,colour=Cluster)) +
  geom_line() +
  theme_bw() + theme(legend.position="none")+
  ggtitle(plotTitle) + ylab("Feature expression") +
  xlab("Time")+
  facet_wrap(~ Cluster)
}
```

```
temp <-list.files(path="Data/", pattern="*.Rdata")
for(i in 1:3){
  clData<-readRDS(paste0("Data/",temp[i]))[[2]]
  nam <- paste("plot", i, sep = "_")
  if(i==1) n="Noise=0.5"
  if(i==2) n="Noise=1.5"
  if(i==3) n="Noise=3"
  assign(nam, plotFunction(clData, plotTitle = n))
}

P<-plot_1+plot_2+plot_3+ plot_layout(guides = "collect")
P
```

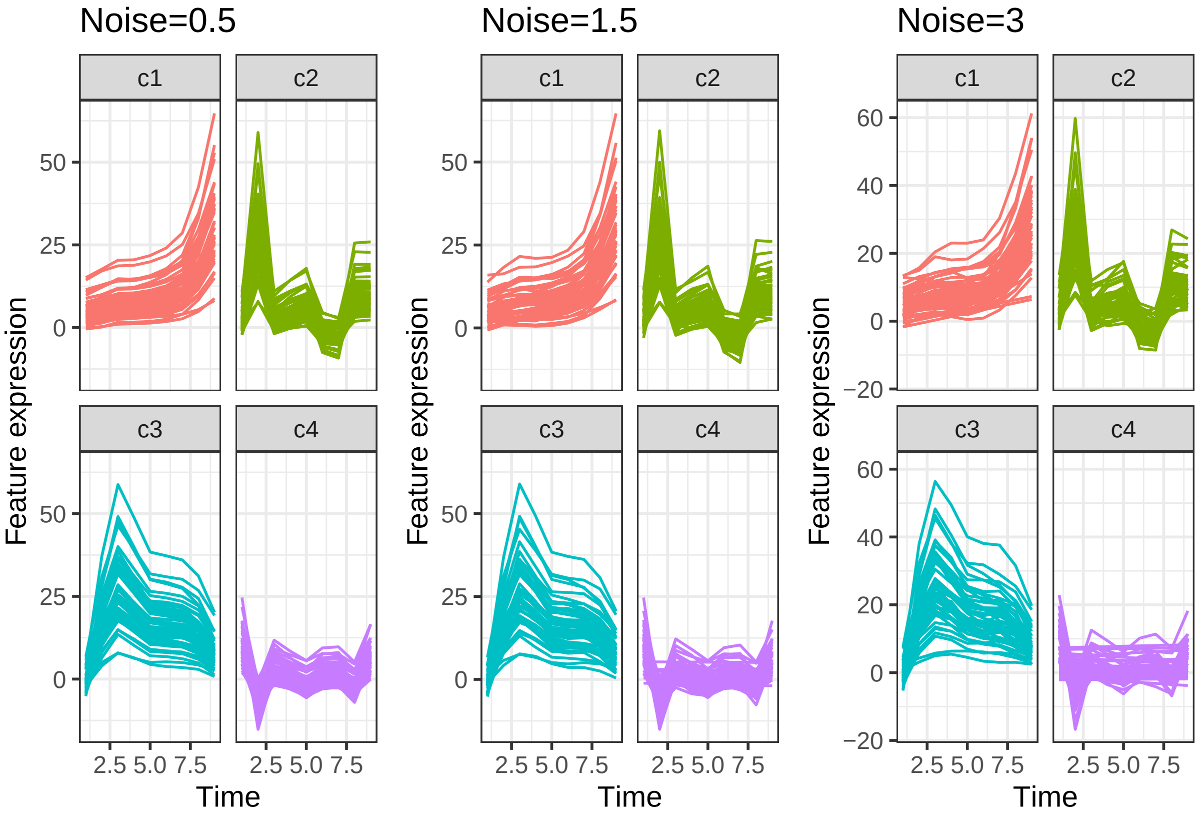

Figure 1: Figure 1, Some simulated LMM profiles with noise levels 0.5, 1.5 and 3

References

Straube, Jasmin, Alain-Dominique Gorse, Proof Centre of Excellence Team, Bevan Emma Huang, and Kim-Anh Lê Cao. 2015. "A Linear Mixed Model Spline Framework for Analysing Time Course 'Omics' Data." Journal Article. *PLOS ONE* 10 (8): e0134540. https://doi.org/10.1371/journal.pone.0134540.